

# Distributed Multi-Robot Cooperation for Information Gathering under Communication Constraints

Alberto Viseras<sup>1</sup>, Zhe Xu<sup>2</sup> and Luis Merino<sup>3</sup>

**Abstract**—Many recent works have proposed algorithms for information gathering that benefit from multi-robot cooperation. However, most algorithms either employ discretization of the state and action spaces, which makes them computationally intractable for robotic systems with complex dynamics; or cannot deal with inter-robot restrictions like e.g. communication constraints. This paper presents an approach for multi-robot information gathering that tackles the two aforementioned issues. To this end we propose an algorithm that combines in an innovative manner Gaussian processes (GPs) to model the physical process of interest, RRT planners to plan paths in a continuous domain, and a distributed decision-making algorithm to achieve multi-robot cooperation. Specifically, we employ the Max-sum algorithm for distributed multi-robot cooperation by defining an information-theoretic utility function together with a path clustering approach. This function maximizes information gathering, subject to inter-robot communication constraints. We validate the proposed approach in simulations, and in a field experiment where three quadcopters explore a simulated wind field. Results demonstrate the effectiveness of the approach.

## I. INTRODUCTION

Information gathering is a fundamental task in a wide range of robotic applications such as environmental monitoring [1], magnetic field intensity mapping [2], traffic modeling [3], or wind field mapping [4]. The objective is to gather information efficiently, saving resources like time, battery, etc. This may be economically advantageous or even life-critical in search and rescue missions.

The information gathering task can clearly benefit from multi-robot cooperation, both in terms of efficiency and robustness. Multi-robot cooperation for information gathering is a topic that has attracted lots of interest, and many strategies have been proposed recently. Here we follow an approach that is widely used in the literature: first, an underlying model of the physical process of interest together with some information metric is employed to predict the impact of robots actions. Second, an active sensing strategy is used to maximize the information metric. Specifically, in this paper we propose a novel approach that employs GPs as underlying model [5], mutual information (MI) as information metric, and an RRT-based [6] nonmyopic sensing strategy.

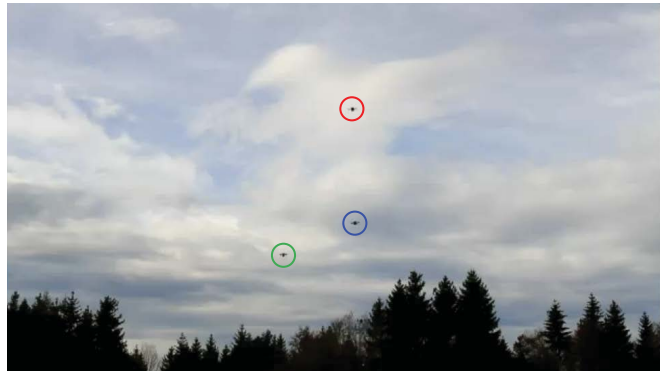


Fig. 1: Three aerial robots exploring a simulated wind field with the distributed information gathering algorithm proposed in this paper.

The works in [7], [4], [8] present multi-robot exploration algorithms that require a central instance. We propose in this paper a distributed cooperation strategy. In contrast to a centralized algorithm, a distributed strategy offers a higher scalability with respect to the number of robots, and is robust under single point failure.

A fundamental aspect in decentralized information gathering is multi-robot cooperation; i.e. how robots should cooperate to maximize the information gathered by the fleet. Several works have proposed decentralized cooperation methods for information gathering with GPs [9], [1], [3], [2]. For example, in [9] the authors employ the Max-sum algorithm [10] to achieve cooperation between multiple robots. Max-sum is a decentralized message-passing algorithm that allows the incorporation of a wide range of utility functions. Here we also propose the use of Max-sum for the multi-robot coordination.

Techniques [9], [1], [3], [2] require a discretization on the robot state and action spaces, and employ graph-based planning algorithms. The works in [11], [12] go one step further and plan in a continuous space. However, [11], [12] do not take into account the robots' dynamics. This does not allow a generalization of those techniques to robots with differential constraints like e.g. fixed-wing aircrafts. Therefore we propose the use of nonmyopic RRT-based planners that perform an exploration in a continuous space, which allows us to generalize our algorithm to a larger class of robots.

There exist multiple single-robot information gathering algorithms that plan in a continuous space [13], [14], [15]. However, the current literature lacks from a decentralized

<sup>1</sup>A. Viseras is with the Institute of Communications and Navigation of the German Aerospace Center (DLR), Oberpfaffenhofen, 82234, Wessling, Germany [alberto.viserasruiz@dlr.de](mailto:alberto.viserasruiz@dlr.de)

<sup>2</sup>Z. Xu is with the Australian Centre for Field Robotics (ACFR), Sydney, Australia, [z.xu@acfr.usyd.edu.au](mailto:z.xu@acfr.usyd.edu.au)

<sup>3</sup>L. Merino is with the Universidad Pablo de Olavide, Crta. Utrera km 1, 41013, Seville, Spain, [lmercab@upo.es](mailto:lmercab@upo.es)

The work of L.M. was partially supported by MINECO (Spain) grant OCELLIMAV (TEC-61708-EXP).

multi-robot cooperation algorithm for information gathering that (i) plans in a continuous space, and (ii) handles mission-specific inter-robot constraints. For example, [13] offers a multi-robot extension with similar characteristics. However, it is designed for tracking applications, and cannot handle constraints.

To the best of our knowledge the closest work is the one proposed in [16]. In [16] the authors propose an algorithm for online decentralized information gathering with spatial-temporal constraints. They employ the augmented Lagrangian method for multi-robot cooperation, which requires constraints and objective functions to be differentiable. This limits the applicability of [16] to a selected class of objective functions and robots. On the contrary, we present a method that is able to incorporate a large class of objective functions and spatial-temporal constraints. We only require that a global objective function must be expressed as a sum of individual robot's objective functions.

To summarize: this paper advances the state of the art by proposing a novel distributed nonmyopic multi-robot cooperation algorithm for information gathering that allows us (i) to consider a large class of robots that are subject to differential constraints, avoiding discretization of robot's state space; (ii) to incorporate spatial and temporal inter-robot constraints, and information-theoretic utilities; (iii) to solve the cooperation problem only with local communication between robots.

We evaluate the proposed approach for mapping of a simulated wind field with multiple aerial vehicles subject to communication constraints. To this end we carried out outdoor experiments with a fleet of three aerial robots (see Figure 1), and simulations with up to four fixed-wing aircrafts.

The paper is structured as follows. First, we state our problem formally in Section II. Then we present our proposed solution in Section III. This is followed in Sections IV, V by an evaluation of the algorithm both in simulations and experiments. To finalize, Section VI closes the paper with conclusions and future work.

## II. PROBLEM STATEMENT

We wish to explore an *a priori* unknown physical process with  $N$  cooperative robots as accurately as possible, in the sense of minimizing the root mean squared error (RMSE) between a process estimate (given by a GPs model) and the (unknown) ground truth. Our goal is to perform an exploration that is efficient given the available resources. To this end, we devise in this paper movement strategies so as to reduce the model's uncertainty over the exploration space as efficiently as possible. The reduction of the model's uncertainty is performed in this work through the maximization of mutual information. This has been showed to be effective to reduce the RMSE between a process estimate and the (unknown) ground truth [17].

Additionally, our problem is subject to the following physical constraints:

- 1) The robot  $i$  motion model is given by  $\mathbf{x}_i(t + dt) = \mathbf{f}(\mathbf{x}_i(t), \mathbf{u}_i(t))$  that relates a robot's current position  $\mathbf{x}_i(t)$  and future position  $\mathbf{x}_i(t + dt)$  given a control input  $\mathbf{u}_i(t)$ .
- 2) Robots can only communicate if they are neighbors, i.e. if they are separated less than a distance  $r_c$ .

Furthermore, we consider a mission-related constraint:

- 3) Communication constraints: a network of robots requires a periodic connectivity, with a maximum disconnection time of  $kdt$  seconds. A communication network at iteration  $k$ , denoted by graph  $\mathcal{G}_c(\mathcal{V}_k, \mathcal{E}_k)$  with vertices  $\mathcal{V}_k$  and edges  $\mathcal{E}_k$ , is connected iff each pair of robots can directly or indirectly communicate. We remark that periodic connectivity allows communication between robots and, therefore, permits multi-robot cooperation. Moreover, it is a requirement in a large class of information gathering applications as indicated in [16].

Let us now introduce some notation that we will use in the remainder of the paper. The position of robot  $i$  will be denoted by  $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^3$ , where  $\mathcal{X}$  corresponds to the free space in the robot's configuration space. The physical process at  $\mathbf{x} \in \mathcal{X}$  is given by  $y(\mathbf{x}) \in \mathbb{R}$ . Typically, however, the process value is not observed directly, but measured using some sensors. Here we assume a simple sensor model that represents a measured process as  $z(\mathbf{x}) = y(\mathbf{x}) + \epsilon(\mathbf{x})$ , where  $z(\mathbf{x})$  is a scalar measurement taken by the robot at position  $\mathbf{x}$  and  $\epsilon(\mathbf{x}) \sim \mathcal{N}(0, \sigma_n^2)$  is a random noise.

We have described an infinite horizon information gathering problem that is subject to constraints. Specifically, we solve the stated information gathering problem sequentially for a finite horizon. In short: first we solve the problem for the next  $k$  iterations, with  $k$  the maximum number of iterations we allow the network to be disconnected. That is, we aim to find the combination of robots' paths  $\mathcal{X}_1, \dots, \mathcal{X}_N$  that maximize a global information-theoretic utility function  $U(\mathcal{X}_1, \dots, \mathcal{X}_N)$ . Then robots will follow the calculated paths and repeat the procedure again. More formally, we aim to solve the following problem:

$$\begin{aligned} & \underset{\mathcal{X}_1, \dots, \mathcal{X}_N}{\text{maximize}} && U(\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_i, \dots, \mathcal{X}_N) \\ & \text{subject to} && \mathcal{X}_i = \{\mathbf{x}_i(t), \mathbf{x}_i(t + dt), \dots, \mathbf{x}_i(t + kdt)\}, \\ & && \mathbf{x}_i(t + (k_t + 1)dt) = \mathbf{f}(\mathbf{x}_i(t + k_t dt), \\ & && \quad \mathbf{u}_i(t + k_t dt)), \\ & && \mathcal{G}_c(\mathcal{V}_k, \mathcal{E}_k) \text{ is connected,} \end{aligned} \tag{1}$$

where  $\mathcal{V}_k = \{\mathbf{x}_1(t + kdt), \dots, \mathbf{x}_N(t + kdt)\}$ , and  $\mathcal{E}_k = \{(\mathbf{x}_i(t + kdt), \mathbf{x}_j(t + kdt))\}$  for all  $i, j \in [1 : N], i \neq j$ , and  $\|\mathbf{x}_i(t + kdt) - \mathbf{x}_j(t + kdt)\|_2 \leq r_c$ .

## III. DISTRIBUTED MULTI-ROBOT EXPLORATION

We describe in this section the algorithm proposed to solve the problem described in Section II. In Section III-A we present an overview of the algorithm. This is followed in Section III-B by a detailed description of each of the subsystems that compose our algorithm.

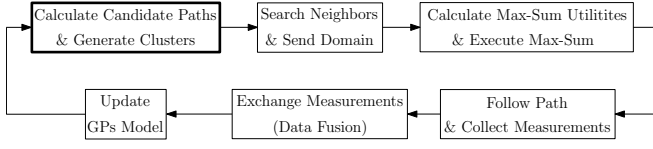


Fig. 2: Algorithm block diagram.

### A. Algorithm Overview

We present a block diagram that describes the algorithm's execution in Figure 2. This algorithm is executed locally by each robot, and works sequentially, where each full iteration corresponds to solving problem (1). The algorithm works as follows: first, each of the robots individually grows an RRT [6] with its current position as root. Then robots send the set of paths contained in the tree to their neighbors in order to cooperate about the optimal assignment of paths. Since the number of paths in the tree could grow indefinitely, we propose a clustering method that reduces the computational complexity of the cooperation procedure by clustering the set of paths of each of the robots (Section III-B.1). Next robots send these clusters to their neighbors (Section III-B.2).

Once a robot receives the clusters, it starts executing a decentralized combinatorial optimization algorithm (Max-sum [10]) to select the cluster that maximizes a user-defined global utility function that is subject to constraints stated in problem (1). Here we define this function as an information-theoretic function – MI – calculated from the underlying GPs model [5].

Once robots find their own cluster that optimizes (1), they communicate the selected cluster to their neighbors. Then, each robot selects a path within its cluster by evaluating a MI (Section III-B.4).

Next robots exchange the gathered measurements through the network; i.e. they perform data fusion (Section III-B.5). These measurements are then employed by the robots to update its GPs model (Section III-B.6). Next we describe each of the aforementioned subsystems in detail.

### B. Algorithm Subsystems

#### 1) Calculate Candidate Paths and Generate Clusters:

The first step of the algorithm is calculating a set of paths that meet the constraints from (1). This is realized with an RRT algorithm, which allows us to incorporate robots with differential constraints.

Ideally, we would like robots to exchange their paths and calculate the combination of paths that yields the highest  $U(\cdot)$ . However this is computationally intractable, as we pointed out in Section III-A. Therefore we introduce the concept of spatial-temporal clusters.

First, robot  $i$  defines  $c_t$  temporal horizons. For example, a path that needs  $5s$  to be traversed by a robot has a temporal horizon of  $5s$ . Then, for each  $c_{t_i} \in [1, 2, \dots, c_t]$ , robot  $i$  extract the RRT paths that have a temporal horizon  $c_{t_i}$ . Next it groups paths of equal  $c_{t_i}$  into  $c_s$  spatial clusters. This last step is realized with the k-means technique. The clustering procedure is depicted in Figure 3.

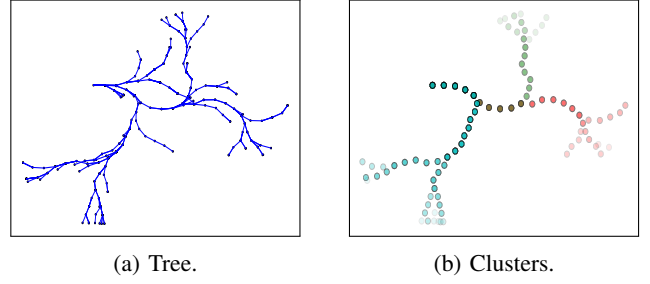


Fig. 3: Example illustrating the clustering process. Fig. 3a depicts an RRT, and Fig. 3b shows the clusters (differentiated by color) calculated with our proposed clustering procedure.

Each robot  $i$  runs the proposed clustering algorithm, which returns  $k_t \times k_s$  clusters.  $C_i = \{C_i^{[1]}, C_i^{[2]}, \dots, C_i^{[k_t k_s]}\}$  denotes the set of clusters of robot  $i$ . Specifically, each cluster  $C_i^{[j]}$  corresponds to a random variable that represents the physical process we aim to explore at positions contained in  $C_i^{[j]}$ .

2) *Search Neighbors and Exchange Domains:* Next, robots search for neighbors since in “Calculate Candidate Paths and Generate Clusters” each robot acts independently. The neighbors search is done by sending an identification message with the robot's ID. Then each robot  $i$  sends to neighbors its set of clusters  $C_i$  that will be exploited by robots for the cooperation step, which we explain next.

3) *Calculate Robot Utilities and Execute Max-Sum:* Once robots receive clusters from their neighbors they cooperate with Max-sum to perform an optimal assignment of clusters to robots. We realize it with the following utility function:

$$U(C_1, \dots, C_N) = f_I(C_1, \dots, C_N) - f_P(C_1, \dots, C_N), \quad (2)$$

where  $f_I(\cdot)$  quantifies the informativeness of a particular assignment of clusters, and  $f_P(\cdot)$  is a penalization factor that takes a positive high value if constraints from (1) are not met.

We define  $f_I(\cdot)$  as the MI between (i) a random variable  $Y_{\mathcal{X}}$  that represents the physical process at a set of positions uniformly sampled from  $\mathcal{X}$ , and (ii) the joint assignation of clusters  $C_1, C_2, \dots, C_N$ ; conditioned on the measurements positions  $\mathbf{M}$  gathered up to now. That is,  $f_I(C_1, \dots, C_N) = I(Y_{\mathcal{X}}; C_1, C_2, \dots, C_N | \mathbf{M})$ .

Our goal is to solve (2) in a decentralized fashion with the Max-sum algorithm. To this end, we must first express  $f_I(\cdot)$  as a sum over the number of robots [10]. We realize it with the chain rule for the MI, and subsequent decomposition of MI as a difference of conditional entropies:

$$\begin{aligned} f_I(C_1, C_2, \dots, C_N) &= I(Y_{\mathcal{X}}; C_1, C_2, \dots, C_N | \mathbf{M}) = \\ &= \sum_{i=1}^N I(Y_{\mathcal{X}}; C_i | C_{i+1}, \dots, C_N, \mathbf{M}) = \\ &= \sum_{i=1}^N H(C_i | C_{i+1}, \dots, C_N, \mathbf{M}) - \\ &\quad - H(C_i | C_{i+1}, \dots, C_N, \mathbf{M}, Y_{\mathcal{X}}). \end{aligned} \quad (3)$$

We have decomposed in (3) a global MI as a sum of MI of individual robots. However, this formulation, as robots only have information about their neighbors, cannot be directly applied for a system that relies on local communication between robots. We solve this issue by applying the principle of *locality* [9], [1], which allows us to assume that two random variables  $C_i, C_j$  are uncorrelated if they correspond to two robots that are not in direct communication. Let us clarify that the *locality* assumption holds for a large class of applications; e.g. in the motivating problem of mapping a wind field. In this case, the structures (thermals) are only a few hundred meters in size, and the characteristic length-scales tend to be shorter. In contrast, the UAV communication distances tend to be in the order of kilometers due to practical considerations.

By considering the *locality* assumption we can now express (2) as:

$$U(C_1, C_2, \dots, C_N) = \sum_{i=1}^N H(C_i | \mathcal{N}(C_{i+1:N}), \mathbf{M}) - H(C_i | \mathcal{N}(C_{i+1:N}), \mathbf{M}, Y_{\mathcal{X}}) - f_P(C_i, \mathcal{N}(C_i)), \quad (4)$$

where  $\mathcal{N}(C_{i+1:N})$  denotes the cluster assignment of the neighbors of robot  $i$  with a higher ID, and  $\mathcal{N}(C_i)$  are the clusters of the neighbors of the  $i$ -th robot.

The penalty function  $f_P(C_i, \mathcal{N}(C_i))$  is also only evaluated between robots that are neighbors. In particular,  $f_P(\cdot)$  is defined as zero if robots are in a configuration that is far from violating the constraints, and as a very high positive value that increases as the robots get closer to a configuration where constraints could be violated.

$f_P(\cdot)$  is composed by two terms related to constraint 3 from Section II. First, we add a network connectivity term, which takes a high value if the resulting network at the end of the robots trajectories gets disconnected. As in [16], we enforce connectivity by guaranteeing that robots form a minimal topology – chain topology; i.e. each robot must be at least connected to the robots with the immediate lower and higher IDs. This way, we meet the communication constraint only with local communications. Second, we add a second term that guarantees a periodic connectivity of the network by letting robots agree about a common temporal horizon.

The formulation of (4) allows robots to solve (1) in a decentralized fashion with the Max-sum algorithm, where Max-sum utilities correspond to each of the individual terms of the sum in (4). The output of the algorithm is a selection of clusters for all robots. Then, each robot must calculate a path  $\mathcal{P}_i$  within the selected cluster. This is done by calculating the mutual information between  $Y_{\mathcal{X}}$ , and all possible path assignments within the selected cluster. We condition the mutual information on the knowledge about the selection of the clusters of neighboring robots, and  $\mathbf{M}$ .

**4) Follow Path and Collect Measurements:** Next, robot  $i$  will follow  $\mathcal{P}_i$ , and will collect measurements  $\mathbf{z}(\mathbf{x}_i)$ , with  $\mathbf{x}_i \in \mathcal{P}_i$ , at positions  $\mathbf{M}$ .

**5) Exchange Measurements (Data Fusion):** Data fusion allows robots to have a common understanding about the

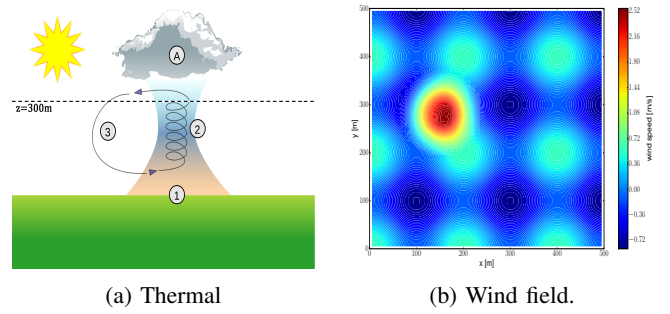


Fig. 4: Illustration of a thermal (4a), and a two dimensional wind field to be explored (4b).

process of interest. In this paper we focus on multi-robot cooperation, and consider decentralized data fusion out of the scope of this work. Therefore, we implement a simple flooding algorithm to carry out the data fusion. Nevertheless, decentralized data fusion approaches like e.g. [3] could be considered.

**6) Update GPs Model:** The last step of the algorithm is updating the GPs model with the newly acquired measurements. Specifically, each of the robots individually perform the update by optimizing the so-called GPs log-marginal likelihood [5].

## IV. SIMULATIONS AND DISCUSSION OF RESULTS

### A. Simulations Setup

**1) Process to Explore: Two-Dimensional Wind Field:** We first validate our algorithm in simulation for the exploration of the vertical component of a wind field (see Figure 4a<sup>1</sup>) with multiple robots. The wind field, depicted in Fig. 4b, corresponds to the model proposed by [18] plus a sinusoidal component in both  $x$  and  $y$  directions. This corresponds to a  $500 \times 500 \text{ m}^2$  two dimensional slice at  $300 \text{ m}$  of a three dimensional wind field.

**2) Robot Model: Simplified Aircraft:** We base our aircraft model on [19]. In contrast to [19] we restrict the aircraft to a constant height, and do not consider the influence of the wind in the aircraft's motion. Although these simplifications are still far from a realistic model, our model allows us to demonstrate the effectiveness of the proposed exploration approach.

Given these assumptions, our aircraft model is defined by:

$$\mathbf{x}(t + dt) = \mathbf{x}(t) + V_{in}(t) dt \quad (5)$$

$$\psi(t + dt) = \psi(t) + \dot{\psi}(t) dt, \quad (6)$$

with  $\mathbf{x}(t)$  the aircraft's position,  $V_{in}(t)$  the aircraft's inertial velocity, and  $\psi$  the heading angle. For airspeed  $V$ , commanded flight path angle  $\theta$ , and commanded bank angle  $\phi$

<sup>1</sup>By Duke (Self-made illustration) [CC BY 2.5 (<http://creativecommons.org/licenses/by/2.5>)], via Wikimedia Commons.



the components of the velocity  $V_{in}(t)$  and  $\dot{\psi}(t)$  are given by:

$$v_x = V \cos \theta \cos \psi \quad (7)$$

$$v_y = V \cos \theta \sin \psi \quad (8)$$

$$\dot{\psi} = \frac{g}{V} \tan(\phi). \quad (9)$$

Here we have considered an aircraft defined by the following parameters:  $dt = 0.5$  s,  $V = 15$  m/s,  $g = 9.8$  m/s<sup>2</sup>,  $\theta = 0$  (constant height),  $\phi \in [-\pi/5, \pi/5]$  rad.

3) *Algorithm Parameters*: We consider four aircrafts to explore the wind field. The aircrafts motion can be approximated by the model described in Section IV-A.2. We define a communication range of  $r_c = 200$  m. For the simulations we run RRT for 1000 iterations, and Max-sum for 5 seconds. We consider four temporal horizons at 2, 5, 7, 10 s, and three spatial divisions.

To the best of our knowledge there are no algorithms in the literature that solve problem (1). Therefore we select three strategies that allow us to benchmark our proposed approach both in terms of the cooperation capabilities to perform the exploration task, and the ability to meet the problem specific constraints, namely network connectivity constraints. These are the following:

- **RandomWalk**: robots move independently following a random path, constrained by the robot motion, generated with RRT. This approach neither aims to meet the constraints nor exchange measurements with the rest of the team.
- **Algorithm**: robots execute the algorithm described in Section III that solves problem (1).
- **AlgorithmNoConstraints**: robots run the algorithm described in Section III with  $f_P(\cdot) = 0$ ; i.e. they do not aim to meet constraints from (1).

We have run Montecarlo simulations to test our approach for systems of one, two, three and four robots. The initial position of the robots is fixed and it is 250 m at the  $y$  direction, and 100, 200, 300, 400 m for the  $x$  direction for robot 1, 2, 3, 4, respectively. For each of the algorithms we average over 100 simulation runs. The algorithm is implemented in Python. We use the robot operating system (ROS) to simulate the algorithm in a distributed fashion.

## B. Simulations Results

1) *Performance of the Exploration Task*: First we evaluate the performance of our proposed algorithm for information gathering, without taking into account any constraints. Therefore we compare in this section RandomWalk and AlgorithmNoConstraints strategies. Specifically, we evaluate the reduction of the root mean squared error (RMSE), calculated between our estimate and ground truth, after a 300 s exploration run.

We compute the RMSE with respect to a set of uniformly distributed points defined over  $\mathcal{X}$  with a spatial resolution of 10 m. We use these points to compare the difference between our estimate, which is the result of GPs regression given the measurements gathered by the robots, and the ground truth. Let us remark that RandomWalk does not perform any

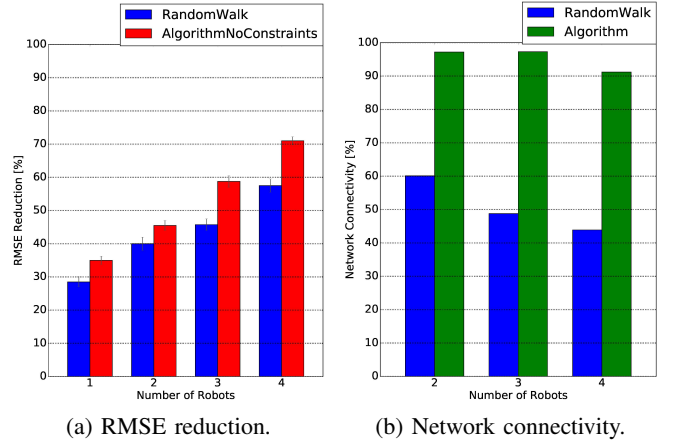


Fig. 5: Algorithm performance in simulations.

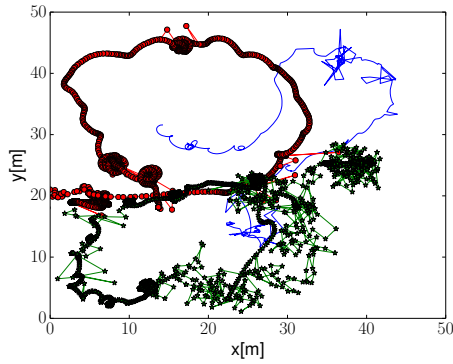
data fusion, which implies that each of the robots only has the measurements taken by themselves. So, in order to obtain a fair comparison with AlgorithmNoConstraints, which fuses data online, we perform a data fusion during post processing for RandomWalk.

We depict in Figure 5a the RMSE reduction for one, two, three and four robots. First fact that we observe is that the performance of AlgorithmNoConstraints increases as we add more robots to the team, which demonstrates the cooperation capabilities of our proposed algorithm. Also AlgorithmNoConstraints offers an increase of performance with respect to RandomWalk of a 7% with one robot, and increases up to a 14% with four robots.

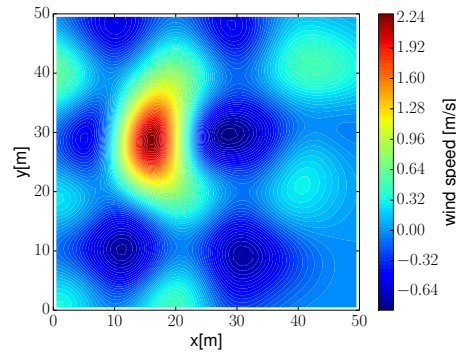
2) *Network Connectivity*: To finalize, we evaluate the ability of the algorithm to meet constraints from (1). We measure the network connectivity as the percentage of the total number of iterations in which the team of robots meet the periodic connectivity constraint. Results are depicted in Fig. 5b. We observe that our proposed algorithm achieves a network connectivity that ranges between 91% and 98%. In contrast, the connectivity resulting from RandomWalk ranges between 40% and 60%.

## V. EXPERIMENTS AND DISCUSSION OF RESULTS

We carried out a field experiment with three quadcopters, which emulate a fixed-wing aircraft dynamics, with the goal of mapping a simulated wind field (Fig. 4b). In Fig. 6 we depict the resulting quadcopters' trajectories, and the wind field estimated at the end of the information gathering task. Experimental results demonstrate that the algorithm proposed in this paper is robust against inaccuracies in positioning solutions, and is able to estimate the wind field with a high accuracy. Specifically, we achieve a 300% RMSE improvement with three robots respect to one, which indicates an efficient multi-robot cooperation. In Figure 1 we show a picture of three quadcopters flying during the experiment's exploration run.



(a) Quadcopters trajectories.



(b) Estimated wind field.

Fig. 6: Results of a field experiment.

## VI. CONCLUSIONS AND FUTURE WORK

The paper presents an approach for multi-robot exploration. The approach extends the state of the art by being able to take into account the motion constraints of the robots, as well as team constraints like e.g. communication restrictions.

The approach has been tested in simulation in the exploration of a wind field. In addition, we have carried out a field experiment with three quadcopters. Results show how the cooperation allows for a more efficient exploration, more evident when the number of robots grow. Furthermore, we have demonstrated how the proposed approach can handle constraints that are relevant for real multi-robot scenarios.

As future work, we plan to apply the approach for the autonomous soaring of gliders, extending the models to a full 3D exploration, and combining the exploration techniques presented with the exploitation of the wind information for longer endurance of the flight.

## REFERENCES

- [1] R. Ouyang, K. H. Low, J. Chen, and P. Jaillet, "Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena," in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 573–580.
- [2] A. Viseras, T. Wiedemann, C. Manss, L. Magel, J. Mueller, D. Shutin, and L. Merino, "Decentralized multi-agent exploration with online-learning of gaussian processes," in *2016 IEEE International*

- Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4222–4229.
- [3] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian Process Decentralized Data Fusion and Active Sensing for Spatiotemporal Traffic Modeling and Prediction in Mobility-on-Demand Systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, 2015.
- [4] D. Cho, J. Ha, S. Lee, S. Moon, and H. Choi, "Informative path planning and mapping with multiple uavs in wind fields," in *Proceedings of the 13th International Symposium on Distributed Autonomous Robotic Systems, DARS, 2016*. [Online]. Available: <http://arxiv.org/abs/1610.01303>
- [5] C. E. Rasmussen and C. K. Williams, *Gaussian Processes for Machine Learning* (Adaptive Computation and Machine Learning). The MIT Press, 2005.
- [6] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
- [7] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient Informative Sensing using Multiple Robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009. [Online]. Available: <http://www.aaai.org/Papers/JAIR/Vol34/JAIR-3420.pdf>
- [8] K. Ma, Z. Ma, L. Liu, and G. S. Sukhatme, "Multi-robot informative and adaptive planning for persistent environmental monitoring," in *Proceedings of the 13th International Symposium on Distributed Autonomous Robotic Systems, DARS, 2016*.
- [9] R. Stranders, A. Farinelli, A. Rogers, and N. R. Jennings, "Decentralised coordination of mobile sensors using the max-sum algorithm," in *Proceedings of the 21st international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., 2009, pp. 299–304.
- [10] A. Farinelli, A. Rogers, A. Petcu, and N. R. Jennings, "Decentralised coordination of low-power embedded devices using the max-sum algorithm," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 2*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 639–646.
- [11] Y. Xu, J. Choi, and S. Oh, "Mobile sensor network navigation using gaussian processes with truncated observations," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1118–1131, 2011.
- [12] M. Jadalila and J. Choi, "Environmental monitoring using autonomous aquatic robots: Sampling algorithms and experiments," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 3, pp. 899–905, 2013.
- [13] D. Levine, B. Luders, and J. P. How, "Information-rich Path Planning with General Constraints using Rapidly-exploring Random Trees," *AIAA Infotech@Aerospace Conference*, no. April, pp. 0–19, 2012.
- [14] G. A. Hollinger and G. S. Sukhatme, "Sampling-based robotic information gathering algorithms," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1271–1287, 2014.
- [15] A. Viseras, D. Shutin, and L. Merino, "Online information gathering using sampling-based planners and gps: an information theoretic approach," in *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*. IEEE, 2017.
- [16] S. K. Gan, R. Fitch, and S. Sukkarieh, "Online decentralized information gathering with spatial-temporal constraints," *Autonomous Robots*, vol. 37, no. 1, pp. 1–25, 2014.
- [17] B. J. Julian, M. Angermann, M. Schwager, and D. Rus, "Distributed robotic sensor networks: An information-theoretic approach," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1134–1154, 2012.
- [18] M. J. Allen, "Updraft model for development of autonomous soaring uninhabited air vehicles," in *Forty Fourth AIAA Aerospace Sciences Meeting and Exhibit, 2006*, pp. 1–19.
- [19] A. Renzaglia, C. Reymann, and S. Lacroix, "Monitoring the evolution of clouds with uavs," in *IEEE International Conference on Robotics and Automation*, 2016.